



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/803,514	03/08/2001	Sridhar Obilisetty	026507-000300US	6110
20350 7590 12/31/2008 TOWNSEND AND TOWNSEND AND CREW, LLP TWO EMBARCADERO CENTER EIGHTH FLOOR SAN FRANCISCO, CA 94111-3834				
EXAMINER				
VU, TUAN A				
ART UNIT		PAPER NUMBER		
2193				
MAIL DATE		DELIVERY MODE		
12/31/2008		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/803,514

Applicant(s)

OBILUSETTY, SRIDHAR

Examiner

TUAN A. VU

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 October 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-6 and 8-25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-6, 8-25 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 10/14/08.

As indicated in Applicant's response, claims 1, 13, 25 have been amended. Claims 1-6, 8-25 are pending in the office action.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1-6, 8-25 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Specifically, claims 1, 13 recite steps of **a)** receiving a plurality of text files defining a component of said application; **b)** executing a program resident on said client system wherein said program include interacting with a server in accordance with 'checking ... periodically for updated versions of text files', 'receiving ... any updated versions of text files' and 'compiling automatically and periodically ...'. Clearly the step of receiving text files occurs prior to executing a resident program to receive periodically text files updates by interaction with a server. According to the Disclosure, the resident agent 205 after being installed, downloads files from the server, and based on the downloaded XML files, the application can be formed (Specifications: pg. 16-17; Fig. 3A,B). Receiving *text files* is disclosed solely by communication with a server via agent 205 instance, such instance being installed once at the client computer

(Specs: pg. 18). The Specifications does not support the understanding that this resident program is JUST for achieving *checking periodically* update files, *receiving periodic updates*, and *compiling updated versions* on a periodic basis; all of which disassociate from receiving text files as in step a). Thus, executing as in b) of a resident program 205 is not subsequent to receiving a plurality of text files recited in a), rendering the executing step b) not truly disclosed as subsequent to step a); but rather is prior to step a). That is, step b) is not clearly supported by the Specifications for including interaction with the server and periodic checking when step a) already entails communications with a server. Therefore, it is deemed that the execution step of a *resident program* as sequenced in the claim is not enabled by the Specifications; i.e. the inventor is not in possession of this step b) subsequent to step a). For the sake of prosecution, the steps b) and a) are merged as one single step in order to allow downloaded XML files to be assembled for the target application (see Specifications, pg. 17-18) with the understanding that executing of an instance of resident program is prior to step a). That is, no weight is given to step a) as a stand alone step prior to executing this resident program.

Claims 2-6, 8-12, 14-24 do not cure to the lack of proper description as identified above, hence are equally rejected for failing to comply with the enablement requirement.

Claim 25 recites b) 'installing a program on said computer system ... **for interacting with a server** for checking ... periodically ... receiving ... any updated versions ... and compiling ... a combination of said updated versions' such that this b) is distinct from or subsequent to step a) installing of text files. The providing of text files is disclosed in the Specifications as server-based (emphasis added) download of XML files via executing a resident agent instance (Fig. 3A,B). The Disclosure does not support installing of a resident program

after installing of text files, because reception of text files cannot happen before execution of a installed executable instance of agent 205 (Specs: pg. 16-18). The inventor is deemed not in possession of step b) for interacting with a server in a way that step b) is subsequent to the installing in step a). Based on the lack of proper support, one cannot make use of the invention as claimed, and step b) and step a) are treated as one single step.

Claim 25 is also rejected for being compliant with the requirement for a proper description, as set forth above.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-6, 8-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bloch et al, USPN: 2002/0129129 (hereinafter Bloch) and further in view of Murray et al, USPN: 6,874,143 (hereinafter Murray).

As per claim 1, Bloch discloses a method for implementing an application on a client computer system, said method comprising steps of:

a) receiving at said client computer system a plurality of text files (e.g. para 0061, pg. 7; step 70-72, Fig. 5; para 0081-0082, pg. 9) wherein each of said text files defines a component of said application;

b) executing a program resident on said client computer system (*AVM 221* - Fig. 2; para 0068-0069, pg. 8; *GUI frame ... waits for the user* - para 0057-0058, pg. 6 – Note: installed and

initialized AVM -- para 0037, pg. 4; para 0050, pg. 5; Fig. 2-3 -- by means of text script files and providing GUI rendered components for user events to take place reads on executing program resident on client)

said resident program comprising instructions for interacting with a server in accordance with receiving automatically versions of text files to create the application (e.g. AVM downloads the XML files - para 0061, pg. 7; step 70-72, Fig. 5; para 0081-0082, pg. 9 -- refer to USC 112, 1st paragraph), and compiling automatically using a combination of versions of said text files to create said application (e.g. Fig. 2, 5; para 0044, pg. 4)

wherein said application is stored on said client computer in a compiled form (para 00100, pg. 11; para 0102, pg. 11; para 0103, pg. 11; para 0105, pg. 12) so as to be executable independent from said program in a runtime environment *independent from the resident program interacting with the server* (e.g. para 0037, pg. 4 -- Note: AVM assembled applications and deployed/executed -- see para 0044, pg. 5 -- reads on being independent from resident program interacting with server in step b, i.e. independent from the server-interaction in terms of receiving of text files, or compiling into executable code)

executing said application in a runtime environment (e.g. para 0037, pg. 4; para 0044, pg. 5) independent from said program and independent from the resident program interacting with the server (Note: AVM assembled applications and deployed/executed -- see para 0044, pg. 5 -- reads on being independent from resident program interacting with server in step b, i.e. independent from the server-interaction in terms of receiving of text files, or compiling into executable code).

Bloch does not explicitly disclose executing said resident program comprises (1) checking automatically and periodically for updated versions of said text files; (2) receiving automatically any updated versions of said text files in response to said program checking for said updated versions when said updated versions are available; then (3) compiling periodically using said combination of said updated versions.

Bloch discloses compiling automatically using a combination of versions of said text files to create said application (e.g. Fig. 2, 5; para 0044, pg. 4) wherein said application is stored on said client computer in a compiled form (para 00100, pg. 11; para 0102, pg. 11; para 0103, pg. 11; para 0105, pg. 12 - Note: executing a script with underlying invoking of procedures reads on compiled form being stored); and further discloses upgrades and fixes and possibility to make use of most recent versions (*upgrades, fixes* - pg. 3, para 0032; *most recent ...version* - pg. 12, para 0107) and addresses the urge for providing latest set of files in accordance to appropriate version of script file or virtual machine files with regard to version (e.g. *receive upgrades as soon as possible* -para 0051-0053, pg. 5-6); hence has taught checking of files and their latest upgrade.

Based on XML header or HTTP text files and browser (Fig. 5; para 0030, pg. 3), it is noted that a file version being included in the header of XML or HTTP files is implicitly disclosed owing to known browser/markup language technologies at the time the invention was made. **Murray**, in a similar framework to develop browser applications with the versions of XML or browser files, discloses a delivery of extension files (e.g. Fig. 15-16), with developer executing a client-side developing tool in contact with server environment to periodically check for download of latest upgrades for XML-implemented files (e.g. Fig. 14; *periodically polling* -

col. 18 line 40 to col. 19 line 17). In light of the desirability of updating browser files to meet the appropriate virtual machine or execution environment version as mentioned above along with the implicitly automated version/format compliance of files processed by a browser engine, it would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance Bloch's desire for proper version of files (para 0053, pg. 6) so that using the browser engine utilizes an automatic upgrade-checking mechanism via **periodic** inquiry or polling as taught in Murray in order for the latest upgraded XML or browser files to be retrieved for compilation (in a corresponding **periodic** scheme) as endeavored in Bloch's use of XML files, whereby extending/developing client or developer's applications. One would be motivated to do so because this periodic remote check/upgrade would enable the executing environment to be provided with the appropriate files according to the version expected for such environment as endeavored by both Bloch's approach and Murray's browser extension tool from above.

As per claim 2, Bloch discloses XML format (Fig. 1, 2, 4,5).

As per claim 3, Bloch discloses server central source for managing and distributing applications or modifications for applications (e.g. *upgrades, fixes* - pg. 3, para 0032; pg. 5, para 0045-0046; pg. 12, para 0109).

As per claim 4, refer to claim 3 and Bloch's Fig. 1, 2, 4,5.

As per claim 5, Bloch discloses executing an application, sending a request and executing the application in parallel while waiting for response from the request (e.g. ... *reports to the Application Handler 302, ... periodically updates* -- pg. 9, para 0080 – 0082 – Note: resolving a URL with data retrieval while leaving the GUI window on for being updated on tree

events changes and notified of download status is equivalent to executing application while waiting for remote response)

As per claim 6, Bloch discloses connectionless application execution (e.g. pg. 8, para 0069; pg. 12, para 0108)

As per claim 8, Bloch discloses modifying application by using a newer text files replacing older files (*upgrades, fixes* - pg. 3, para 0032; *most recent ...version* - pg. 12, para 0107).

As per claim 9, Bloch discloses graphical user interface (e.g. Fig. 6).

As per claim 10, Bloch discloses application being communication preferences for database invocation (e.g. pg. 7, para 0063; Preference Handler 303 - Fig. 4)

As per claim 11, Bloch discloses data management application (e.g. step 52 – Fig. 5; Manager 301 -Fig. 4 - Note: downloading files to assemble manager module reads on application being a management application).

As per claim 12, Bloch discloses component being part of logic of application (pg. 1, para 0012; pg. 4, para 0037).

As per claim 13, Bloch discloses a computer system comprising: a bus; a computer-readable memory unit coupled to said bus; and a processor coupled to said bus, said processor for executing a method for implementing an application comprising:

a)receiving at said computer system a plurality of text files wherein each of said text files defines a component of said application (refer to claim 1);

b)executing a program resident on said computer system (refer to claim 1), wherein said program comprises instructions for interacting with a server in accordance with receiving of

said text files (Note: receiving text files and receiving by resident program treated as one – see USC § 112, 1st rejection); and compiling automatically using a combination of versions of said text files to create said application (refer to claim 1), wherein said application is stored on said computer system in a compiled form so as to be executable independent from said program in a runtime environment independent from the resident program interacting with the server (refer to claim 1); and

executing said application on said runtime environment, independent from said program in a runtime environment independent from said resident program interacting with the server (refer to claim 1).

Bloch does not explicitly disclose execution of resident program for (1) checking automatically and *periodically for updated versions* of said text files; (2) receiving automatically *any updated versions of said text files* in response to said program checking for said updated versions when said updated versions are available; (3) *compiling periodically using said combination of said updated versions*. But these limitations have been addressed in claim 1.

As per claims 14-18, 20-24, these claims correspond to claims 2-6, 8-12 respectively, hence are rejected with the corresponding rejections as set forth therein, respectively.

As per claim 19, Bloch discloses text files particular to client system (e.g. pg. 4, para 0037; pg. 5, para 0047, 0050)

As per claim 25, Bloch discloses a computer-usable medium having computer-readable program code embodied therein for causing a computer system to perform a method comprising code for:

a) installing on said computer system a plurality of text files (refer to claim 1) wherein each of said text files defines a component of said application;

b) installing a program on said computer system (para 0057, pg 6), wherein said program comprises instructions for interacting with a server in accordance with receiving automatically versions of text files to create the application; and compiling automatically using a combination of versions of said text files to create said application program (refer to claim 1 – Note: receiving text files and receiving by resident program treated as one – see USC 112 rejection), wherein said application is stored on said computer system in a compiled form so as to be executable independent from said program (refer to claim 1), and

executing said application on said runtime independent from said program in a runtime environment independent from said program and independent from said program interacting with the server (refer to claim 1).

Bloch does not explicitly disclose resident program for (1) checking automatically and periodically for updated versions of said text files; (2) receiving automatically any updated versions of said text files in response to said program checking for said updated versions when said updated versions are available; (3) compiling automatically and periodically using combination of said updated versions of said text files. But these limitations have been addressed in claim 1.

Nor does Bloch explicitly disclose created application being stored in compiled form so as to be executable independent from said program in a runtime environment independent from said program. But this limitation has been rendered obvious as set forth in claim 1.

Response to Arguments

6. Applicant's arguments filed 10/16/08 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 103 Rejection:

(A) Applicants have submitted that the Bloch does not disclose executing application on said runtime independent said (resident) program based on the "on the fly" approach as cited (Appl. Rmrks pg. 8, 4th para). Previously, the rejection has established a rationale as to why this application after being compiled based on the text files via instance of a AVM would be an obvious executable that can be run on runtime environment, in light of the combined teachings as set forth in the 103 Rationale. The previously rationale of 103 included and addressed the obvious concept regarding destiny/purport of compiled code being available for future execution and use outside of its compiling environment; but the changes in the claim language necessitate herein reconsideration of the previous use of Bloch.

The language of the claim now recites 'independent from said program ... program interacting with the server'; the program being one recited in step b). Interpretation of this language has triggered another form of rejection (emphasis added), rendering the argument moot. For example, 'on the fly' observation is not commensurate with the very specifics of the Office Action, which is not actually and necessarily based on (Applicant-exhibited) paragraphs 0032, 0034, 0037, 0038, 0047, 0058, 0060 of Bloch.

(B) Applicants have submitted that AVM-based and assembled application being stored temporarily by Bloch and becoming secondary to the AVM cannot persist outside of the lifetime of the AVM or retained as soon as the AVM is removed (Appl. Rmrks, pg 9, top paras). The claim does not describe 'executing said application' in specific details to enforce what 'executing

... independent from' actually consists of. That is, the limitation 'independent from said program ... independent from program interacting with the server' **has been interpreted** as though execution runtime of any of the compiled applications is only respective to the resident program as recited in step b, i.e. resident program interacting with the server (emphasis added) in terms of checking and receiving of versions of files and compiling. In that light, **application code assembled from a compilation instance and deployed as in Bloch does execute independent from the timeframe or memory context in which Bloch's AVM initially communicates with the server to download text files** with interpretation of 'said program' solely based on sub-steps (1) (2) (3) of step b. The deployed applications in Bloch are not dependent from the context in which 'said program' performs (1) (2) and (3). The 'independent from said program interacting with the server' has been deemed fulfilled by the cited portions regarding execution of assembled code in Bloch's AVM. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(C) Applicants have submitted that many portions in Bloch teach that compiled applications in Bloch depend on the AVM 221, and evidences such as Virtual Applications, Task lists, user interactions in para 0010, message boxes in para 0102, RPCs in para 0103, database handler in para 0105, testing in para 0109 cannot constitute execution being 'independent from anything' (Appl. Rmrks bottom pg. 9 to pg. 10-11). The paragraphs proffered by the Applicants are exactly not utilized of the current Office Action addressing the 'independent from' limitation. The language regarding 'independent from said program' has been construed using broad

reasonable interpretation, as analyzed in section B above. Lacking specifics as to how this 'independent from' is all about, one cannot be convinced as to how the deployed applications in Bloch would be dependent from the runtime context of the AVM in light of the interaction with the server exactly as sub-steps (1), (2) or (3) recited in step b. The argument is deemed insufficient in demonstrating why deploying applications in Bloch is necessarily subsumed in the runtime context of receiving XML files or compiling files into deployable code; and this argument is referred back to section B above.

In all, the claims will stand rejected as set forth in the Office Action.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

December 29, 2008